



Exclusión logica #1

Transcripción

[00:00] Bien, vamos a implementar nuestro caso de uso del negocio, que sería el delete lógico. Vamos a ver aquí lo que nos retornan y por ejemplo, ya vimos que tengo el id, pero yo no quiero borrarlo de la base de datos, entonces lo que yo necesito hacer es un filtrado, un filtrado de mis datos para traer los que estén activos.

[00:24] Este caso de uso es muy común que lo llamamos, que le vamos a agregar un flag, una bandera, algún alguna característica a estos médicos que me digan que están activos. ¿Esto qué es? Agregar un campo más a mi identidad médico, que sería ese de aquí. Y por lo tanto, también a mi base de datos, un campo que me diga: “Diego, este médico está activo, este médico no está activo”.

[00:51] Eso le suena a un migration. Entonces, como ya sabemos, detenemos el servidor porque vamos a trabajar con migrations, vamos a hacer un nuevo migration, limpiamos esto y primero que todo voy a crear mi campo aquí y lo voy a llamar por ejemplo un private Boolean, para que sea solo true or false y se va a llamar activo.

[01:13] Tengo mi activo aquí. Y a nivel de constructor, también voy a modificar el constructor. Le voy a decir `this.activo = true`, porque yo llamo al constructor cada que estoy creando este objeto, todo médico que se esté creando por defecto está activo. Entonces ya tengo aquí, en `actualizarDatos` no voy a modificar nada. A nivel de Java ya está todo, no necesito hacer nada más. ¿Qué es lo que sigue ahora?

[01:42] El código SQL. Entonces aquí yo ya tengo un alter table para crear un nuevo campo a mi tabla médicos, voy a copiar lo mismo, pero no voy a copiar esto aquí y le voy a pegar aquí, ya vimos que eso no funciona. Lo que yo voy a hacer es copiar todo esto, el archivo me refiero y lo voy a actualizar aquí a V3__alter-table-medicos-add-activo. Le damos OK, no lo agregamos a Git aún. Y ya está.

[02:18] Aquí lo que yo le voy a decir es alter table médicos add activo y no va a ser un varchar, esto va a ser un tinyint y del tipo no va a ser no nulo, porque si no los parámetros que están anteriormente creados no van a funcionar. Y eso sería todo por ahora. Por ahora tengo mi tinyint, mi tabla médicos y le voy agregando el activo.

[02:51] Ahora si yo quiero filtrarlo por este parámetro yo necesito que los demás parámetros estén actualizados, que los demás registros en la base de datos estén actualizados. Para esto yo lo que yo le voy a decir es update médicos set activo = 1. Porque en lo que él va a hacer es actualizar todos los registros anteriores que ya existen en mi base de datos.

[03:23] Me refiero todos estos que no tienen la columna activo. Como yo ya la estoy agregando, y no quiero que esté vacía, no quiero que esté nula, entonces yo aquí mismo ya voy a actualizar los registros anteriores para que todo esté sincronizado hasta este momento.

[03:40] Voy a ejecutar mi migration, ya tengo mi alter table, tengo mi update, voy a iniciar mi servidor, y vamos a ver qué es lo que me dice Spring. Vemos que está iniciando. Vemos que validó 3 migrations. Vemos que aplicó satisfactoriamente una migración, entonces parece que el migration funcionó correctamente.

[04:06] ¿Y qué más vamos a hacer? Vamos a validar. Abrimos aquí. Listamos nuestra tabla. Y vemos que ahora activo ya se agregó, recuerda que no había antes. Ahora está activo y todo está 1 por defecto, entonces todos mis médicos

ahora ya están activos. Primer paso completo, check. Cerremos esto, ya no lo necesitamos, cerramos el migration.

[04:28] Ya fue ejecutado, ya no lo necesitamos tampoco. Y dado que lo que queremos hacer es un delete lógico, vamos aquí a Controller, dado que lo que queremos es un delete lógico, entonces no queremos llamar al método delete, queremos llamar al método update. ¿Por qué? Vamos a ver aquí.

[04:50] Yo voy a dejar comentado este método solamente para que lo tengan de referencia luego. Y voy a actualizar aquí, primero que todo eso, he obtenido el médico y aquí yo le voy a decir que haga un update, aquí por ejemplo `medico.desactivarMedico`. Entonces, esto como parámetro va a recibir el médico.

[05:29] Listo. Aquí voy a venir, le voy a crear el método y en desactivar médico lo que yo le voy a decir es `medico.activo = false`. Y eso sería todo lo que yo tengo que hacer. Regreso aquí. Y ya no debo hacer nada. ¿Por qué? Porque ya está anotado con `@Transactional`.

[05:55] Entonces cuando termine este método, este médico que ya fue actualizado y el parámetro que hemos actualizado es el nuevo que hemos creado, `activo = false`, entonces, estos datos van a ser commiteados en la base de datos. Aquí yo le voy a decir que este es el delete en base de datos y a este de aquí le vamos a llamar un delete lógico.

[06:23] Entonces vamos a guardar y vamos a ver si es que funciona mi método. Esperamos aquí. Vamos a borrar el médico con el id 5, por ejemplo, ya no existe un 3. Ahora le damos a borrar 5 enviamos, me da 200 OK. Voy a listar mis médicos. Y cuando yo le diga listado, vemos que sigue apareciendo. Aquí sucedió algo, voy a revisar mi base de datos.

[06:53] Voy a venir aquí, y el primer médico vemos que sigue con `activo` igual 1. ¿Esto por qué es?

